IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: Andrea BRAGAGNINI, et al.		Atty. Docket No.:	007511.00003
Serial No.:	10/535,476	Art Unit:	2184
Filed:	May 17, 2005	Examiner:	John B. Roche
For:	METHOD FOR DIRECT MEMORY ACCESS, RELATED ARCHITECTURE AND COMPUTER PROGRAM PRODUCT	Confirmation No.:	3939

APPEAL BRIEF

U.S. Patent and Trademark Office
Customer Service Window
Mail Stop - Appeal
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

This is an Appeal Brief filed in support of Appellant's November 29, 2010, Notice of Appeal. Appeal is taken from the final Office Action mailed August 31, 2010 (hereafter, "the Final Office Action").

Please charge any fees to our Deposit Account No. 19-0733. Also, a one-month extension of time to file this paper is hereby requested.

REAL PARTY IN INTEREST

37 C.F.R. § 41.37(c)(1)(i)

The owner of this application, and the real party in interest, is Telecom Italia SpA.

RELATED APPEALS AND INTERFERENCES

37 C.F.R. § 41.37(c)(1)(ii)

There are no related appeals or interferences.

STATUS OF CLAIMS

37 C.F.R. § 41.37(c)(1)(iii)

Claims 1, 3-9, and 11-15 are pending and rejected. No claims have been canceled or added. Appellant hereby appeals the rejection of claims 1, 3-9, and 11-15.

STATUS OF AMENDMENTS

37 C.F.R. § 41.37(c)(1)(iv)

The most recent amendment to the claims was filed June 16, 2009. No amendments have been made subsequent to the Final Office Action.

SUMMARY OF CLAIMED SUBJECT MATTER

37 C.F.R. § 41.37(c)(1)(v)

In making reference herein to various embodiments in the specification text and drawings to explain the claimed invention, Appellant does not intend to limit the claims to those embodiments; all citations to the specification including drawings are illustrative unless otherwise explicitly stated. ***All citations to the specification are with reference to the clean version of the substitute specification filed February 21, 2008.***

Techniques for direct memory access (DMA) are generally disclosed by Appellant's specification. (Specification, p. 1, lines 11-12). For example, various blocks (e.g., IP blocks) may be associated with respective DMA modules, each including an input buffer and an output buffer. (Specification, p. 3, lines 22-24). These DMA modules may be coupled over a data transfer facility in a chain arrangement of their input and output buffers. (Specification, p. 3, line 25 to p. 4, line 5). Each DMA module may write data to, and read data from, a respective IP block using its output and input buffers. (Specification, p. 4, lines 5-9).

An example of such a DMA architecture is shown in Figs. 2 and 11. In this example, the DMA modules are labeled as IDMA A, IDMA B, and IDMA C; and the blocks are labeled as elements A, B, and C.

The input and output buffers of the DMA modules may be operated in such a way that the writing of data from the input buffer of a DMA module into the respective IP block is started when the input buffer is at least partly filled with data. (Specification, p. 4, lines 9-12). Also,

when reading of data from the IP block into the output buffer of the DMA module is completed, the data in the output buffer may either be transferred to the input buffer of another downstream DMA module, or, if the DMA module is the last in the chain, provided as output data. (Specification, p. 4, lines 12-17).

Independent claim 1 is directed to the following (again, all references to the specification are merely illustrative and are not intended to limit the claims):

A method of exchanging data within a direct memory access arrangement including a plurality of IP blocks (Figs. 2 and 11; elements A, B, and/or C), the method comprising the steps of:

associating with said IP blocks respective DMA modules (Figs. 2 and 11; IDMA A, IDMA B, and/or IDMA C) each including an input buffer (Fig. 11; elements 11A, 11B, and/or 11C) and an output buffer (Fig. 11; elements 12A, 12B, and/or 12C); (specification, p. 3, lines 22-24)

coupling said respective DMA modules over a data transfer facility (Figs. 2 and 11; BUS) in a chain arrangement where each DMA module, other than the last in the chain, has its respective output buffers coupled to the input buffer of another of said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its respective input buffer coupled to the output buffer of another of said DMA modules upstream in the chain; (specification, p. 3 line 25 to p. 4, line 5; p. 23, line 23 to p. 24, line 6)

causing each of said DMA modules to interact with the respective IP block by writing data from the input buffer of the DMA module into the respective IP block and reading data from the respective IP block into the output buffer of the DMA module; and (specification, p. 4, lines 5-9)

operating said input and output buffers in such a way that:

said writing of data from the input buffer of the DMA module into the respective IP block is started when the respective input buffer is at least partly filled with data, and (specification, p. 4, lines 9-12)

when said reading of data from the respective IP block into the output buffer of the DMA module is completed, the data in the output buffer of the DMA module are transferred to the input buffer of the DMA module downstream in the chain or, in the case of the last DMA module in the chain, are provided as output data; (specification, p. 4, lines 12-17)

associating with said output buffers and input buffers coupled in the chain at least one intermediate block (Fig. 11; elements 16A and/or 16B) to control data transfer between said coupled buffers; and (specification, p. 4, lines 19-21)

controlling transfer of data between said coupled buffers over said data transfer facility by issuing at least one request of a requesting buffer for a buffer coupled therewith to indicate at least one transfer condition selected out of the group consisting of:

data existing to be transferred and enough space existing for receiving said data when transferred; (specification, p. 4, line 22 to p. 5, line 1)

issuing at least one corresponding acknowledgment towards said requesting buffer confirming that the said at least one transfer condition is met; and (specification, p. 5, lines 1-3)

transferring data between said requesting buffer and said coupled buffer, whereby said data transfer facility is left free between said at least one request and said at least one acknowledgement. (specification, p. 5, lines 4-7)

Independent claim 5 is directed to the following (again, all references to the specification are merely illustrative and are not intended to limit the claims):

An apparatus, comprising:

a bus (Figs. 2 and 11; BUS); (specification, p. 7, lines 1-3)

a processor (Figs. 2 and 11; CPU) coupled to the bus; and (specification, p. 7, line 3)

a plurality of respective direct memory access (DMA) modules (Figs. 2 and 11; IDMA A, IDMA B, and/or IDMA C) each associated with a different IP block (Figs. 2 and 11; elements A, B, and/or C), the DMA modules being coupled to each other over the bus, each DMA module including: (specification, p. 3, lines 22-24)

an input buffer (Fig. 11; elements 11A, 11B, and/or 11C) configured to write data into a respective one of the IP blocks and exchange data with said bus, and (specification, p. 3, lines 22-24; p. 4, lines 5-9)

an output buffer (Fig. 11; elements 12A, 12B, and/or 12C) configured to read data from said respective IP block and exchange data with said bus, said

DMA modules being coupled together by the bus in a chain so that each of said DMA modules, other than the last in the chain, has its respective output buffer coupled to the input buffer of another of said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its input buffer coupled to the output buffer of another of said DMA modules upstream in the chain. (specification, p. 3, line 22 to p. 4, line 9; p. 23, line 23 to p. 24, line 6)

Independent claim 11 is directed to the following (again, all references to the specification are merely illustrative and are not intended to limit the claims):

An apparatus, comprising:

a bus (Figs. 2 and 11; BUS); (specification, p. 7, lines 1-3)

a processor (Figs. 2 and 11; CPU) coupled to the bus; and (specification, p. 7, line 3)

a plurality of respective direct memory access (DMA) modules (Figs. 2 and 11; IDMA A, IDMA B, and/or IDMA C) each associated with a different circuit block (Figs. 2 and 11; elements A, B, and/or C), the DMA modules being coupled to each other over the bus, each DMA module including: (specification, p. 3, lines 22-24)

a first buffer (Fig. 11; elements 11A, 11B, and/or 11C) configured to write data into a respective one of the circuit blocks and exchange data with said bus, and (specification, p. 3, lines 22-24; p. 4, lines 5-9)

a second buffer (Fig. 11; elements 12A, 12B, and/or 12C) configured to read data from said respective circuit block and exchange data with said bus, said DMA modules being coupled together by the bus in a chain so that each of a plural subset of the DMA modules has its respective second buffer coupled to the first buffer of another of said plurality of DMA modules and each of said plural subset of DMA modules has its respective first buffer coupled to the second buffer of another of said DMA modules. (specification, p. 3, line 22 to p. 4, line 9; p. 23, line 23 to p. 24, line 6)

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

37 C.F.R. § 41.37(c)(1)(vi)

Claims 1, 3-9, and 11-15 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,781,799 to Leger, et al. ("Leger") in view of U.S. publication no. 2003/0033454 A1 to Walker et al. ("Walker"), and further in view of U.S. Patent No. 6,870,929 to Greene ("Greene").

ARGUMENT

37 C.F.R. § 41.37(c)(1)(vii)

A. Independent Claim 1, and Dependent Claims 3 and 4

Independent claim 1 recites:

> coupling said respective DMA modules over a data transfer facility in a chain arrangement where each DMA module, other than the last in the chain, has its respective output buffers coupled to the input buffer of another of said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its respective input buffer coupled to the output buffer of another of said DMA modules upstream in the chain....

and further recites:

> when said reading of data from the respective IP block into the output buffer of the DMA module is completed, the data in the output buffer of the DMA module are transferred to the input buffer of the DMA module downstream in the chain or, in the case of the last DMA module in the chain, are provided as output data....

Leger

The Final Office Action compares the recited DMA modules with DMA controllers 20 of Leger (Fig. 2), alleging that the DMA controllers 20 are connected in a daisy chain. However, as conceded by the Final Office Action, Leger fails to teach or suggest (1) coupling input and

output buffers of DMA modules in a chain, and (2) transferring the recited data from the output buffer of one DMA to the input buffer of another DMA, as recited in claim 1.

Leger plus Walker

Instead, the Final Office Action alleges that it would have been obvious to modify Leger to incorporate input and output buffers that are allegedly part of the multi-port DMA 5 of Walker (Fig. 2) into each of the DMA controllers 20 of Leger, so as to facilitate coupling between DMA modules. The alleged motivation for modifying Leger is to have enabled other modules to access the system bus while a DMA controller is handling a transfer between two modules (referring to Walker at paragraph 5, lines 2-6).

However, Walker at paragraph 5 is not concerned with transferring data between two DMA controllers, as alleged by the Final Office Action, but rather discusses the problems that arise when a DMA controller handles a transfer between two "modules," or locations. See also Walker paragraphs 0009 and 0011, which discuss how a DMA controller can perform a data transfer between two locations, such as a processor, memory, or bus, without occupying the system bus. In this case, there is nothing to teach or suggest that the two modules, or locations, are two DMA controllers that are communicating with each other. In fact, Walker does not even teach or suggest using more than one DMA controller.

The Final Office Action responds to the above argument by stating that "the coupling of DMA ports as shown in Walker *could* be used to ensure the transfer of data between said ports, which could be interpreted as modules." Final Office Action, numbered paragraph 13 (italics added). Appellant respectfully disagrees, for the following reasons.

First, while Walker does disclose connecting any of ports A, B, C, and D together, these ports are all part of the *same* DMA controller. Thus, this never teaches or suggests connecting ports of different DMA controllers together.

Second, it is respectfully submitted that the above-quoted response in the Final Office Action is mere speculation, without support from Walker or any other source. The reasoning of the Final Office Action implies the absurd conclusion that anything *could* be connected to anything else. In fact, Walker does not teach or suggest connecting the ports of the DMA controller to anything other than (1) other ports of that same DMA controller, or (2) other types of modules. There is nothing in Walker to teach, suggest, or even imply chaining together multiple DMA controllers, or that the other modules could be other DMA controllers. Walker does not envision the need for multiple DMA controllers.

Thus, even if Leger were somehow modified with the buffers of Walker as proposed, this still would not result in a system where two DMA controllers are coupled together in the manner claimed.

Appellant further respectfully disagrees with the stated reasoning for modifying Leger with Walker. The Final Office Action states that it would have been obvious to modify Leger with Walker "because the addition of an input buffer and an output buffer can facilitate coupling between two DMA modules." This is a circular argument, because the argument assumes that Leger or Walker teaches coupling between two DMA modules. As previously explained, Walker does not teach or suggest coupling a port of the DMA module with a port of another

DMA module; there is no other DMA module in Walker. Thus, the resulting modified Leger/Walker system still would not have coupled the ports of multiple DMA modules together.

The Final Office Action goes on to state that the motivation for modifying Leger with Walker "would have been to mitigate the typical disadvantage of conventional DMA controllers, wherein said conventional DMA controller acquires complete control of a bus." But this alleged motivation is taken from Walker, which again, does not teach or suggest coupling multiple DMA controllers together. Thus, this alleged reasoning does not explain why or how one would have arrived at DMA modules coupled together as claimed. In fact, Walker alleges being capable of mitigating the bus control issue using only a single DMA controller.

The Final Office Action also alleges that Leger teaches DMA controllers connected in a daisy-chain configuration, and so one could have the initial DMA controller connected to a system bus on one end and to another DMA controller on another. Final Office Action, p. 12. While Leger indeed discloses daisy-chaining DMA controllers together, these are daisy-chained only by a *multi-chip negotiation line* connected between MAckIn and MAckOut connections. Leger, Fig. 2; col. 9, lines 24-31. These MAckIn and MAckOut connections are not the recited input and output buffers (nor does the Final Office Action ever allege that they are). Thus, at best, combining Leger with Walker might arguably have resulted in a collection of DMA controllers having their negotiation lines daisy-chained together, but not their input/output buffers chained in the manner recited in claim 1.

Leger plus Walker, plus Greene

The Final Office Action further relies on Greene for the feature of the DMA modules being coupled in a chain as recited. Greene uses a plurality of cipher stages 802 arranged in a pipeline to perform data encryption. Each cipher stage 802 includes a combinational section 804 and a pipeline register 806 (see Fig. 8). The Final Office Action alleges that it would have been obvious to modify the data-exchanging method of Leger/Walker to include the combinational sections and pipeline registers of Greene, because the chain of such combinational sections and pipeline registers would enhance the overall performance of the data processing transfer. The alleged motivation is to have provided a higher throughput (referring to col. 4, lines 42-43 of Greene).

However, there is no evidence that using a pipelined arrangement in a DMA system in the same manner as the encryption elements are pipelined in Green would have been expected to result in higher throughput, or would even have been expected to work at all. The Final Office Action seems to rely on the assumption that a pipelined system always provides higher throughput than a non-pipelined system. To the contrary, there are many systems in which a parallel architecture might provide higher throughput than a pipelined (serial) architecture. For example, massively parallel-processing computer systems have often been used because of their extremely high throughput. In fact, the addition of the encryption elements of Greene to the Leger/Walker system would, if anything, have had the opposite effect as alleged by the Final Office Action. In fact, if the combined system worked at all, the addition of an encryption function to the DMA must surely result in *lower throughput* than the pure DMA function alone.

Put another way, Greene stands for the proposition that, in solving the *encryption* problem particular to that patent, a pipelined encryption system is efficient. There is no reason to believe from the teachings of Greene that pipelining in a completely different system having a completely different function – encryption versus DMA memory management – would (1) result in a higher throughput system, or (2) operate at all, especially without significant experimentation. In other words, there is neither a motivation nor an expectation of success, based on the teachings of Greene, Leger, or Walker, to have "pipelined" or "daisy chained" (to use the words of the Final Office Action), the input/output buffers of DMA controllers.

And, even though Greene, Walker, and Leger may all be directed to "data processing" (as alleged by the Final Office Action), data processing is an extremely broad category that covers a wide range of technology. In fact, Greene is directed to a very different type of data processing – i.e., memory management versus pipelined encryption, with very different problems and solutions. Following the Final Office Action's arguments and assumptions to their logical conclusion would mean it were obvious to have improved throughput by pipelining *any* elements of *any* data processing system. It is, of course, not that simple, because pipelining often slow processes down, depending upon the situation. This is why, for example, parallel-processing computing systems have long existed.

Moreover, despite the insistence to the contrary at page 14 of the Final Office Action, it appears that the Final Office Action proposes to have added the pipelined elements 804 of the cipher stages 802 of Greene to the Leger/Walker system. Appellant again respectfully submits that, even though DMA and encryption functions both involve the "processing" of data, one

could not have simply transplanted the various pipelined encryption elements from the Greene encryption system into the Leger/Walker DMA system and expected it to work. The combinational sections of Greene (i.e., the encryption components) have nothing to do with the DMA functionality of Leger and Walker. And, the Final Office Action has not cited any valid reason why one would have wanted to add encryption to a DMA system in the first place.

Therefore, it is respectfully submitted that the alleged reasoning for modifying Walker/Leger with Greene – to speed up the Walker/Leger system – is flawed. And, for the reasons discussed above, the proposed modification would not work and would not result in the claimed invention.

Conclusion

For at least these reasons, it is submitted that claim 1 is allowable over Leger, Walker, and Greene, either alone or in combination.

Dependent Claims 3 and 4

Dependent claims 3 and 4 are also not obvious over Leger, Walker, and Greene, at least by virtue of depending from claim 1.

B. Independent Claims 5 and 11, and Dependent Claims 6-9 and 12-15

Independent claim 5 recites:

> a plurality of respective direct memory access (DMA) modules each associated with a different IP block, the DMA modules being coupled to each other over the bus, each DMA module including:
>
> an input buffer configured to write data into a respective one of the IP blocks and exchange data with said bus, and
>
> an output buffer configured to read data from said respective IP block and exchange data with said bus, said DMA modules being coupled together by the bus in a chain so that each of said DMA modules, other than the last in the chain, has its respective output buffer coupled to the input buffer of another of said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its input buffer coupled to the output buffer of another of said DMA modules upstream in the chain.

Leger

The Final Office Action again compares the recited DMA modules with DMA controllers 20 of Leger (Fig. 2), alleging that the DMA controllers 20 are connected in a daisy chain. As apparently conceded by the Final Office Action, Leger fails to teach or suggest "said DMA modules being coupled together by the bus in a chain so that each of said DMA modules, other than the last in the chain, has its respective output buffer coupled to the input buffer of another of

said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its input buffer coupled to the output buffer of another of said DMA modules upstream in the chain," as recited in claim 5.

Leger plus Walker

Instead, the Final Office Action's position that it would have been obvious to modify Leger to incorporate input and output buffers that are allegedly part of the multi-port DMA 5 of Walker (Fig. 2) into each of the DMA controllers 20 of Leger, so as to facilitate coupling between DMA modules. The alleged motivation for modifying Leger would have been to enable other modules to access the system bus while a DMA controller is handling a transfer between two modules (referring to Walker at paragraph 5, lines 2-6).

However, for the reasons discussed above with regard to claim 1, Walker at paragraph 5 is not concerned with transferring data between two DMA controllers, but rather discusses the problems that arise when a DMA controller handles a transfer between two "modules," or locations. See also Walker paragraphs 0009 and 0011, which discuss how a DMA controller can perform a data transfer between two locations, such as a processor, memory, or bus, without occupying the system bus. In this case, there is nothing to teach or suggest that the two modules, or locations, are two DMA controllers that are communicating with each other. In fact, Walker does not even teach or suggest using more than one DMA controller.

As previously explained, while Walker discloses connecting any of ports A, B, C, and D together, these ports are all part of the *same* DMA controller. Thus, Walker never teaches or suggests connecting the ports of different DMA controllers together. Moreover, as discussed

above with regard to claim 1, the reasoning of the Final Office Action implies the absurd conclusion that anything *could* be connected to anything else. In fact, Walker does not teach or suggest connecting the ports of the DMA controller to anything other than (1) other ports of that same DMA controller, or (2) other types of modules. There is nothing in Walker to teach, suggest, or even imply chaining together multiple DMA controllers, or that the other modules could be other DMA controllers. Walker does not envision a need for multiple DMA controllers.

Thus, even if Leger were somehow modified with the buffers of Walker as proposed, this still would not result in a system where two DMA controllers are coupled together in the manner claimed.

Appellant further respectfully disagrees with the stated reasoning for modifying Leger with Walker. The Final Office Action takes the position that it would have been obvious to modify Leger with Walker "because the addition of an input buffer and an output buffer can facilitate coupling between two DMA modules." As discussed above with regard to claim 1, this is a circular argument, because the argument assumes that Leger or Walker teaches coupling between two DMA modules. As previously explained, Walker does not teach or suggest coupling a port of the DMA module with another DMA module; there is no other DMA module in Walker. Thus, the resulting modified Leger/Walker system still would not couple the ports of multiple DMA modules together.

The Final Office Action goes on to argue that the motivation for modifying Leger with Walker "would have been to mitigate the typical disadvantage of conventional DMA controllers, wherein said conventional DMA controller acquires complete control of a bus." However, as

previously discussed with regard to claim 1, this alleged motivation is taken from Walker, which again, does not teach or suggest coupling multiple DMA controllers together. Thus, this alleged reasoning does not explain why or how one would have arrived at DMA modules coupled together in the manner recited in claim 5. In fact, Walker alleges being capable of mitigating the bus control issue using only a single DMA controller.

The Final Office Action also alleges that Leger teaches DMA controllers connected in a daisy-chain configuration, and so one could have the initial DMA controller connected to a system bus on one end and to another DMA controller on another. Final Office Action, p. 12. As previously explained, while Leger indeed discloses daisy-chaining DMA controllers together, these are daisy-chained only by a *multi-chip negotiation line* connected between MAckIn and MAckOut connections. Leger, Fig. 2; col. 9, lines 24-31. These MAckIn and MAckOut connections are not the recited input and output buffers (nor does the Final Office Action ever allege that they are). Thus, at best, combining Leger with Walker might arguably result in a collection of DMA controllers that have their negotiation lines daisy-chained together, but not their input/output buffers configured in the chain recited in claim 5.

Leger plus Walker, plus Greene

The Final Office Action again relies on Greene for the feature of the DMA modules being coupled in a chain as recited. Greene uses a plurality of cipher stages 802 arranged in a pipeline to perform data encryption. Each cipher stage 802 includes a combinational section 804 and a pipeline register 806 (see Fig. 8). The Final Office Action alleges that it would have been obvious to modify the data-exchanging method of Leger/Walker to include the combinational

sections and pipeline registers of Greene, because the chain of such combinational sections and pipeline registers would enhance the overall performance of the data processing transfer. The alleged motivation would have been to provide a higher throughput (referring to col. 4, lines 42-43 of Greene).

However, as discussed above with regard to claim 1, there is no evidence that using a pipelined arrangement in a DMA system in the same manner as the encryption elements are pipelined in Green would have been expected to result in higher throughput, or would even have been expected to work at all.

And, even though Greene, Walker, and Leger may all be directed to "data processing" (as alleged by the Final Office Action), data processing is an extremely broad category that covers a wide range of technology. As discussed above with regard to claim 1, Greene is directed to a very different type of data processing – i.e., memory management versus pipelined encryption, with very different problems and solutions that do not necessarily mesh together.

Therefore, it is respectfully submitted that the alleged reasoning for modifying Walker/Leger with Greene – to speed up the Walker/Leger system – is flawed. And, for the reasons discussed above, the proposed modification would not work and would not result in the claimed invention.

Conclusion

For at least these reasons, it is submitted that independent claim 5 is allowable over Leger, Walker, and Greene, either alone or in combination.

Independent Claim 11

Independent claim 11 is also allowable over Leger, Walker, and Greene for at least similar reasons as discussed above with regard to claim 5.

Dependent Claims 6-9 and 12-15

Dependent claims 6-9 and 12-15 are also not obvious over Leger, Walker, and Greene, at least by virtue of depending from either claim 5 or claim 11.

CONCLUSION

For all of the foregoing reasons, Appellant respectfully submits that the final rejection of claims 1, 3-9, and 11-15 is improper and should be reversed.

Respectfully submitted,
BANNER & WITCOFF, LTD.

Dated: February 10, 2011

By: /Jordan N. Bodner/
Jordan N. Bodner
Registration No. 42,338

1100 13th Street, N.W.
Suite 1200
Washington, D.C. 20005
Tel: (202) 824-3000
Fax: (202) 824-3001

CLAIMS APPENDIX

37 C.F.R. § 41.37(c)(1)(viii)

Claims involved in the appeal:

1. A method of exchanging data within a direct memory access arrangement including a plurality of IP blocks, the method comprising the steps of:

associating with said IP blocks respective DMA modules each including an input buffer and an output buffer;

coupling said respective DMA modules over a data transfer facility in a chain arrangement where each DMA module, other than the last in the chain, has its respective output buffers coupled to the input buffer of another of said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its respective input buffer coupled to the output buffer of another of said DMA modules upstream in the chain;

causing each of said DMA modules to interact with the respective IP block by writing data from the input buffer of the DMA module into the respective IP block and reading data from the respective IP block into the output buffer of the DMA module; and

operating said input and output buffers in such a way that:

said writing of data from the input buffer of the DMA module into the respective IP block is started when the respective input buffer is at least partly filled with data, and

when said reading of data from the respective IP block into the output buffer of the DMA module is completed, the data in the output buffer of the DMA module are transferred to the input buffer of the DMA module downstream in the chain or, in the case of the last DMA module in the chain, are provided as output data;

associating with said output buffers and input buffers coupled in the chain at least one intermediate block to control data transfer between said coupled buffers; and

controlling transfer of data between said coupled buffers over said data transfer facility by issuing at least one request of a requesting buffer for a buffer coupled therewith to indicate at least one transfer condition selected out of the group consisting of:

data existing to be transferred and enough space existing for receiving said data when transferred;

issuing at least one corresponding acknowledgment towards said requesting buffer confirming that the said at least one transfer condition is met; and

transferring data between said requesting buffer and said coupled buffer, whereby said data transfer facility is left free between said at least one request and said at least one acknowledgement.

3. The method of claim 1, further comprising the steps of:

including a CPU in said arrangement;

using said CPU for transferring data to be processed into the input buffer of the first DMA module in said chain; and

using said CPU for collecting said output data from the output buffer of the last DMA module in said chain.

4. The method of claim 3, further comprising the step of configuring said DMA modules via said CPU.

5. An apparatus, comprising:

a bus;

a processor coupled to the bus; and

a plurality of respective direct memory access (DMA) modules each associated with a different IP block, the DMA modules being coupled to each other over the bus, each DMA module including:

an input buffer configured to write data into a respective one of the IP blocks and exchange data with said bus, and

an output buffer configured to read data from said respective IP block and exchange data with said bus, said DMA modules being coupled together by the bus in a chain so that each of said DMA modules, other than the last in the chain, has its respective output buffer coupled to the input buffer of another of said DMA modules downstream in the chain and each of said DMA modules, other than the first in the chain, has its input buffer coupled to the output buffer of another of said DMA modules upstream in the chain.

6. The apparatus of claim 5 wherein at least one of said input and output buffers has a fixed data width with respect to said data transfers facility and a selectively variable data width with respect to said respective IP blocks.

7. The apparatus of claim 5, further comprising a slave interface module configured to read from outside the apparatus data relating to at least one parameter selected from the group consisting of:

a parameter indicating how many bits are available for reading in at least one of said input buffers,

a parameter indicating how many bits are present in at least one of said input buffers,

a parameter indicating how many bits are available for reading in at least one of said output buffers, and

a parameter indicating how many bits are present in at least one of said output buffers.

8. The apparatus of claim 5 further comprising a reprogrammable finite state machine configured to drive operation of said apparatus by receiving data from at least one of said input buffers, downloading data into said respective IP block corresponding to said at least one of said input buffers, receiving data from said respective IP block, and storing data in at least one of said output buffers.

9. The apparatus of claim 5 wherein at least one of said input buffers and output buffers is associated with a respective master block that is configured to exchange data between the associated buffer and said bus, said master block being configured to be coupled in a data exchange relationship to a buffer in a homologous direct memory access module in an arrangement, wherein said master block and said buffer coupled thereto are configured to:

issue at least one request of a requesting buffer for a buffer coupled therewith to indicate at least one transfer condition selected out of the group consisting of data existing to be transferred and enough space existing for receiving said data when transferred;

issue at least one corresponding acknowledgement towards said requesting buffer confirming that the said at least one transfer condition is met; and

transfer data between said requesting buffer and said coupled buffer, whereby said data transfer facility is left free between said at least one request and said at least one acknowledgement.

11. An apparatus, comprising:

a bus;

a processor coupled to the bus; and

a plurality of respective direct memory access (DMA) modules each associated with a different circuit block, the DMA modules being coupled to each other over the bus, each DMA module including:

a first buffer configured to write data into a respective one of the circuit blocks and exchange data with said bus, and

a second buffer configured to read data from said respective circuit block and exchange data with said bus, said DMA modules being coupled together by the bus in a chain so that each of a plural subset of the DMA modules has its respective second buffer coupled to the first buffer of another of said plurality of DMA modules and each of said plural subset of DMA modules has its respective first buffer coupled to the second buffer of another of said DMA modules.

12. The apparatus of claim 11, wherein the plurality of DMA modules comprises three DMA modules.

13. The apparatus of claim 11, wherein at least one of said first and second buffers has a fixed data width with respect to said bus and a selectively variable data width with respect to said respective circuit blocks.

14. The apparatus of claim 11, wherein each of said circuit blocks comprises an IP block.

15. The apparatus of claim 11, wherein said processor comprises a central processing unit (CPU).

EVIDENCE APPENDIX
37 C.F.R. § 41.37(c)(1)(ix)

NONE.

RELATED PROCEEDINGS APPENDIX
37 C.F.R. § 41.37(c)(1)(x)

NONE.